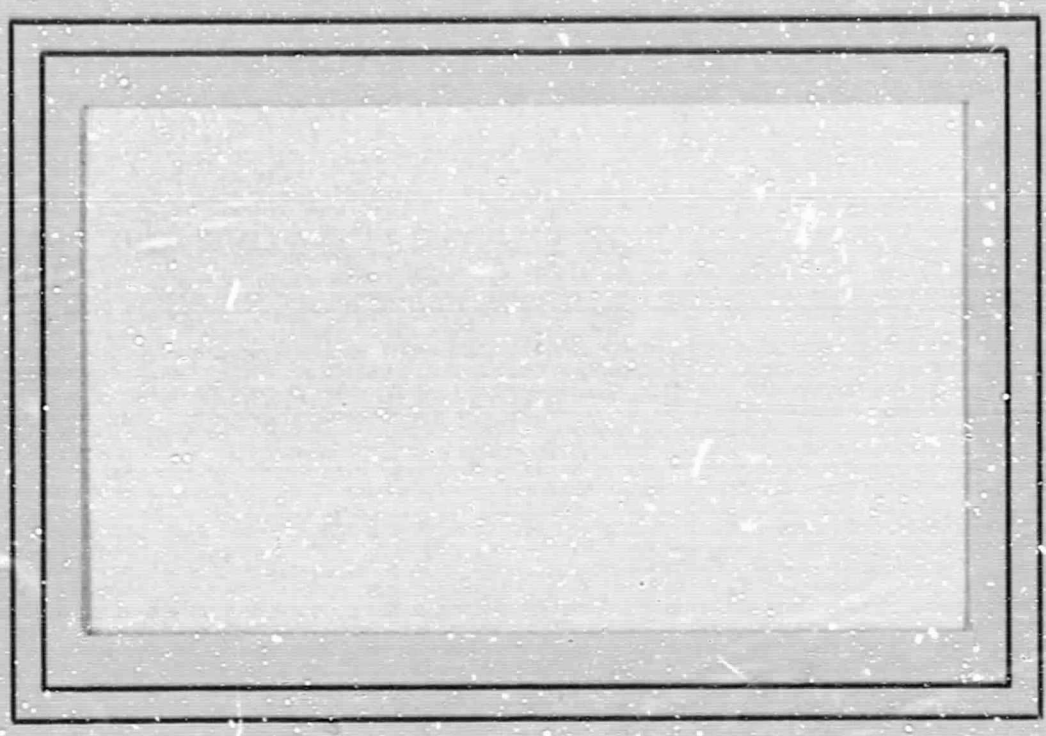

UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

Technical Report 69-89 April 1969
~~NsG 398~~ and GJ-231
NGL-21-002-008

A SYSTEM FOR TESTING ITERATIVE METHODS FOR SOLVING NONLINEAR EQUATIONS

by

Charles K. Mesztenyi and Werner C. Rheinboldt

NGL-21-002-008
This research was supported in part by Grant ~~NsG 398~~ from the National Aeronautics and Space Administration and by Grant GJ-231 from the National Science Foundation to the Computer Science Center of the University of Maryland.

Abstract

This report describes a system of computer programs for testing iterative methods for solving nonlinear systems of equations. The programs are written in FORTRAN V for the Univac 1108. The system can be updated by adding or deleting methods and sets of equations, and it can be used with various sets of initial approximations for the iterations. The results of the iterations are collected in a data file which can be printed in table form for comparing different methods.

Table of Contents

1. Introduction

During the past decade growing attention has been directed toward the development of iterative methods for the numerical solution of systems of nonlinear equations

$$
\begin{aligned}
f_1(x_1,\ldots,x_n) &= 0 \\
f_2(x_1,\ldots,x_n) &= 0 \\
&\ldots \\
f_n(x_1,\ldots,x_n) &= 0
\end{aligned}
\tag{1}
$$

For example, besides the many variations of Newton's method, there are several variations of the secant method, the so-called generalized linear methods, and the large number of minimization methods in case the f_i's are the partial derivatives of a nonlinear functional. For a survey of these methods we refer to the forthcoming book by J. M. Ortega and W. C. Rheinboldt*.

None of these methods is satisfactory for all types of systems, and, in fact, for different systems the behavior of a certain method may change drastically. In addition, the convergence behavior of any method depends, in most cases very critically, on the choice of the initial approximation(s). Although there is a growing literature about convergence theorems for the different methods, the conditions of many of these theorems for a specific system are frequently not numerically checkable or they predict only very small convergence domains. Moreover, only a few results are known about the stability behavior of the various methods under round-off.

---

*Iterative Solution of n-Dimensional Nonlinear Equations, Blaisdell Publishing Company, 1970.

More than in many other areas of numerical analysis, it is therefore of considerable importance to conduct computational experiments with various methods, in each case, for many different choices of systems and for a large number of initial approximations. A series of such experiments with one given method may give some insight into the behavior of that method under different conditions. Usually, it is more important to have an experimental comparison of different methods under varying conditions.

Such a test program will quickly run into considerable data-handling difficulties unless from the very outset a flexible integrated system of programs has been devised which allows the incorporation of new methods, systems and initial approximations at any time, and which permits the accumulation of test results in a form suitable for future processing.

This report describes such a system of programs for testing any number of iterative methods for solving any number of systems of equations (1) under arbitrary initial conditions. In the next chapter, we describe the adopted numerical tests for the convergence behavior of an iterative method. The third chapter describes the various files of the system, and the fourth chapter, the test program. The last two chapters give details of the presently available postprocessing programs.

2. Numerical tests for convergence

Let

(2) $$x^0, x^1, \ldots, x^{i-1}, x^i$$

be the sequence of iterates obtained from some process after i steps, and

(3) $$F_0, F_1, \ldots, F_{i-1}, F_i$$

the corresponding function values. Here

$$x^j = (x_1^j, x_2^j, \ldots, x_n^j)^T, \quad F_j = (f_1(x^j), f_2(x^j), \ldots, f_n(x^j))^T$$

are n-dimensional vectors with the norms $\|x^j\|$ and $\|F_j\|$.

The convergence tests are to be incorporated in each different method program and consist of tests performed after every step of the iteration to determine whether the process should terminate, and of tests performed on the last-obtained iterate after termination.

The tests for the termination of an iterative process after the i-th step are as follows:

a) If the method breaks down during the i-th step, e.g., if a singular system of equations arises, then the iteration is terminated and the termination symbol B is generated.

b) If $\|x^i\| \geq 10^{20}$ or $\|F_i\| \geq 10^{20}$, the iteration is terminated. The iteration is considered to be diverging, and the termination symbol D is used.

c) If $\|x^i - x^{i-1}\| \leq \varepsilon_2$, the iteration is terminated; it is considered to be converging, and the termination symbol C is generated.

d) If

$$\|x^i - x^{i-1}\| < \|x^{i-1} - x^{i-2}\| < \ldots < \|x^{i-i_0+1} - x^{i-i_0}\|$$

and $(i \geq i_0)$

$$\|x^i - x^{i-1}\| \leq \varepsilon_3 \|x^i\| ,$$

then the iteration is terminated; it is considered to be converging, and again the symbol C is generated.

e) If

$$\|x^i - x^{i-1}\| > \|x^{i-1} - x^{i-2}\| > \ldots \quad \|x^{i-i_0+1} - x^{i-i_0}\| , \quad (i \geq i_0)$$

then the iteration is terminated; it is considered to be diverging; this is again denoted by D.

f) If

$$\|F_i\| > \|F_{i-1}\| > \ldots > \|F_{i-i_0+1}\|$$

and $(i \geq i_0)$

$$\|x^i - x^{i-1}\| \geq \|x^{i-1} - x^{i-2}\| ,$$

then the iteration is terminated; it is considered to be diverging and also indicated by D.

g) If $i > MAX$, the iteration is terminated; it is considered to be indecisive, and the termination symbol I is generated.

The second group of tests are performed when the termination symbol was either B (case a) or C (case c and d). If

$$\|F_i\| \leq \varepsilon_1$$

then B is changed to BC. If

$$\| F_i \| > \varepsilon_1$$

then C is changed to CB. BC indicates an instability of the method near the solution. CB indicates convergence to a point which is not a solution of (1) or very slow convergence.

The parameters, MAX, ε_1, ε_2, ε_3 and i_0, are input parameters; they can be changed for different iterations.

3. File organization

The Method-file is the collection of various iterative methods for solving any nonlinear system of equations. Each method has a unique name consisting of not more than six characters.

The Function-file (or F-file) is the collection of nonlinear systems of equations of the form (1). These systems of equations are sequentially numbered in the file, and one specific system of equations is specified by the letter F and its sequence number.

The Initial-value-file (or S-file) is the collection of sets of initial approximations for the iterative processes. These sets are sequentially numbered in the file, and k-S refers to the k-th set of initial approximations. Each set of initial approximations consists of one or several points in R^n. These are again numbered sequentially, and k-S-i refers to the i-th point in the set k-S.

The PR-file is the program file. All programs associated with the Test System were written in FORTRAN V* for the Univac 1108. The symbolic and relocatable forms of these programs are collected on this tape-file including the programs associated with the previously-defined files. The file may be updated by adding or deleting method subroutines, function routines, initial-value generating subroutines, or with any auxiliary programs. Using the collector of EXEC 8 for the Univac 1108, any subprogram can be selected from the PR-file for execution. The following rules for naming programs must be applied when the file is updated:

---

*Except a small assembly language routine which gives the elapsed time in msec.

	Symbolic	Relocatable
Method subroutine	A.METHOD/'NAME'	A.METHDX/'NAME'
Initial value generating subroutine	A.S/n	A.SX/n
Function routine	A.F/m	A.FX/m
	A.D/m	A.DX/m

where 'NAME' is the unique name of the method, and n and m are the appropriate sequence numbers. Further description of these routines can be found in 4.5 - 4.7.

At the present, the PR-file contains three main programs:

i. Method Testing Program

ii. List of Tape Output File

iii. Table Generating Program

The Method Testing Program, consisting of several subprograms, tests one method by solving one system of nonlinear equations with one set of initial approximations. The List of Tape Output File lists the contents of File 2 (see below). The Table Generating Program collects data from File 2 and prints it out in table form.

The Output-File is the collection of the outputs of the Method Testing Program. It consists of printed, punched and tape outputs. The generated and updated tape output is called File-2.

4. Method Testing Program

4.1 Deck setup

For the execution of the Method Testing Program the following card deck is necessary:

```
Initial Control Cards:
'RUN        ...
'ASG,T      PR,8C,t1
'ASG,T      2,8C,t2
'ASG,T      A,F
'COPIN,SR   PR.
'FREE       PR.

Cards to define one run:
'MAP,I      A.PROC,A.PROCX
        SEG     PRO
        IN      A.MAINX,A.PCHX,A.PRNTX,A.GAUSX,A.SCALX,A.TIMEX
        IN      A.LTWOX
        IN      A.METHDX/'name'
        IN      A.FX/n,A.DX/n
        IN      A.SX/m
'XQT        A.PROCX
    (one data card)

Terminating cards:
'FREE       2.
'FREE       A.
'FIN
```

Initial Control Cards: These cards define the files; t1 and t2 should be replaced by the tape reel numbers of the files PR and 2.

The "Cards to define one run" may be repeated before the "Terminating Cards" to execute more than one run with different combinations of methods, functions, etc. The 'IN' cards define the selection of the necessary programs. The first 'IN' card defines the main program and subroutines needed for any run. The second 'IN' card defines the subroutine which calculates the norm of a vector. The third 'IN' card selects the method used for this run; 'name' should be replaced by the alpha-numeric name of the method. The fourth and fifth 'IN' cards select the system of equations and the initial value sets; 'n' and 'm' should be replaced by the corresponding file indices.

The "Data Card" has the following format:

col. 1 - 5 MAX = maximum number of iterations allowed

10 NPR = print index
NPR = 1 step-by-step print during iteration
= 0 only final results to be printed

14 - 15 NPU = Punch and Tape index
NPU = 0 or 2 no punch
= 1 or 3 punch final results
= 0 or 1 no tape output
= 2 or 3 final results written on File 2
negative sign indicates a new tape;
positive sign indicates that the tape
must be positioned.

16 - 25 ε_1

26 - 35 ε_2

36 - 45 ε_3

46 - 55 i_0

} input parameters described in Chapter 2

(Format: 3I5, 4E10.5)

Example:

```
50    0    -2    .1E-6    .1E-6    .1E-5    .5E1
```

4.2 Available subroutines

The following subroutines are incorporated in the main program package; they may be used in any new method subroutine with the following calling sequences:

VMPV (N,A,IA,B,IB) is a function routine for computing the scalar product of the N-dimensional vectors A and B. IA and IB are the index increments for calculating the locations of the components of A and B.

$$\text{VMPV (N,A,IA,B,IB)} = \sum_{i=1}^{N} A(1+(i-1)*IA)*B(1+(i-1)*IB)$$

The routine calculates the products and the summation in double precision arithmetic and truncates the result to single precision.

GINV(*,A,B,N) is a matrix inversion subroutine. The N × N matrix A is inverted by Gauss elimination with pivoting and the inverse is placed in B. Both matrices A and B should be dimensioned 25 × 25. The input matrix A remains unchanged unless B refers to the same storage, which is permitted. The internal calculation is done in double precision arithmetic; * is the error return when the matrix A has been found numerically singular.

GSLV (*,E,F,N) is a subroutine for solving the linear system of equations

$$\sum_{j=1}^{N} E(I,J)*X(J) = F(I)$$

The solution X replaces F and the matrix E remains unchanged; E must be dimensioned 25 × 25. The routine solves the equations by Gauss elimination with pivoting in double precision arithmetic and gives truncated final results. The return to * occurs if the matrix E is numerically singular.

4.3 Outputs

4.3.1 - Printed output

The first page of the output is a heading page. It consists of the data supplied by the data card, and, of the initial value and function file identifications. An example of this heading is given on page A-1.

Following the heading page, each iteration processed with an "Initial-Value" is printed as follows:

(a) The "Initial Value" with its index numbers in the given set S/m

(b) Step-by-step results of the iteration consisting of the step-number, the maximum norm of the remainder, and the independent variable. If the print index NPR is zero on the data card, this printing is suppressed.

(c) Results of the iteration and the file identifications. The results of the iteration consist of the termination symbol, the solution index, the number of function evaluations, and the time in msec spent in the method program. As described in Chapter 2, the termination return can be one of the following:

C = Convergence return from the method subroutine with X as a last iterate and either

$$\|X - Z_i\| \leq \|Z_i\| \cdot \varepsilon_3$$

or

$$\|F(x)\| \leq \varepsilon_1$$

where Z_i is one of the solutions supplied by the function routine and the norm is defined by the selected norm routine. In the first case the solution index is set to i; if the first condition is not satisfied while the second one is, then the solution index is set to zero.

CB = Convergence return from the method subroutine with X as a last iterate but with neither one of the above conditions satisfied.

D = Divergence return from the method subroutine.

I = After the allowed maximum number of iterations neither convergence nor divergence was established by the method subroutine

B = The method broke down and the last obtained point X did not satisfy either one of the conditions under 'C'.

BC = The method broke down but the last obtained point X satisfied one or both of the conditions under 'C'.

The solution index is set to i ≠ 0 whenever the last obtained point is close to the i-th solution supplied by the function routine, that is, when the first condition of 'C' is satisfied. Otherwise, it is always set to zero.

An example for the front page is shown on page A-1; examples for the iterations are shown on pages A-2 and A-3.

4.3.2 Punched output

When punch is requested by the data card, one summary card is punched for each iteration with the following format:

1.	col. 2		Identifier number = 4
2.	"	4 - 9	= Name of the method
3.	"	10 - 14	= File index of the Function routine
	"	15	= Letter F
4.	"	16 - 20	= File index of the Initial Value Generating routine; m
	"	21	= Letter S
5.	"	22 - 24	= Initial value index in the set S/m
6.	"	27 - 28	= Type of return
7.	"	29 - 31	= Solution index
8.	"	32 - 37	= Number of executed iteration steps
9.	"	38 - 43	= Number of function evaluations
10.	"	44 - 49	= Time spent in msec
11.	"	51 - 56	= Norm used
12.	"	57 - 60	= Maximum number of iteration steps allowed

13. " 61 - 66 = ε_1
14. " 67 - 72 = ε_2
15. " 73 - 78 = ε_3
16. " 79 - 80 = i_0

} in E6.1 formats

Page A-4 shows an example using a list of output cards.

4.3.3 Tape output (File 2)

When tape output is requested by the data card, one logical binary record of 26 words is written on tape (File 2) for each iteration. The first 16 words are the 16 data listed in the Punched output; the last 10 words are reserved for possible other classifications or remarks. A dummy-record, with 26 words of zeros, is used to mark the end of the file.

4.4 Norm-Function Routines

The Method Testing System provides two norm routines:

$$\text{L-TWO:} \quad \|x\| = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}$$

$$\text{L-MAX:} \quad \|x\| = \text{Max} \, (|x_i| \, , \; i=1,\ldots,N)$$

but it is possible to insert other norm-function routines. A properly programmed Norm routine is illustrated on page A-5. The first entry FNORM (X,N), (line 2), is used to return the value of the norm of the N-dimensional vector X. The second entry FNORMH (FF), (line 16), is used to return the alphabetic name of the norm; FF is a dummy argument.

4.5 Method Routines

The Testing Program requires the Method routines to be programmed in a standard form. This form is illustrated by the Newton Method on page A-6:

Lines 1 - 14, initialization part, and lines 28 - 49, testing part, are standard.

The first line is a Fortran control statement. Note that the name of the method appears as a version.

The arguments of the routine are as follows (line 2):

*1	Return position if divergence is detected
*2	Return position if no divergence or convergence after MAX number of iteration steps
*3	Return position if the method breaks down. In the example, this return occurs when the Newton method tries to solve a numerically singular linear system (lines 22 and 52)
N	Dimension
X(i)	i = 1,...,N. Initial value vector, which is replaced by the iterated values
MAX	Maximum number of iterative steps allowed
ACC(1)	ε_1. "zero" for the norm of the dependent variable, F(X)
ACC(2)	ε_2. "zero" for the norm of the independent variable, X
ACC(3)	ε_3. relative accuracy required for X
ACC(4)	i_0, in floating point format, number of consecutive terms required to detect convergence or divergence
IT	Number of iterations performed
AN	Name of the method

In lines 8 - 11, the counters are set to zero; IT = iterative step, ICC = convergence counter, IDC1 = divergence counter for X, IDC2 = divergence counter for F(X).

In lines 12 - 14, i_0 as an integer is established, the output routine is called with the initial value and the name of the method is placed into the argument.

Lines 15 - 27 are dependent upon the particular method, except for line 20, where the iteration loop starts.

The testing part (lines 28 - 49) assumes the newly-obtained independent variable X_{IT} in X and the last correction $(X_{IT} - X_{IT-1})$ or $(X_{IT-1} - X_{IT})$ in DX. First the Output routine is called, then the norms are calculated. If the norm of the last correction is less than ε_2, a normal (C) return occurs. To avoid machine overflow, a divergence return occurs if the norm of X or F(X) is greater than or equal to 10^{20} (line 33). Lines 35 - 40 update the counters except at the first iteration step. Lines 42 - 46 test for convergence or divergence; they are skipped in the first i_0 steps. Lines 47 and 48 save the norms for the next iteration step, and line 49 tests the total number of iterations.

The special comment lines (16, 27, 50) indicate where statements particular to the method may be inserted.

4.6 Function Routines

A properly programmed function routine is illustrated on page A-7.

The Fortran control lines, 1 and 19, contain the File index number of the function as version (1 in this example).

The first entry FXK(K,X), line 2, is used to calculate the remainder of the k-th equation at the given point X, and the counter NF for the number of function evaluations is incremented.

The second entry, NFEK(K), line 13, gives the number of function evaluations and resets the counter; K is a dummy argument. NF and FN are made equivalent to avoid floating point-integer conversion.

The third entry, FX(F,X), line 20, gives the remainders of all equations at the given point X.

The fourth entry, DERIVE(X,DF), line 26, gives the Jacobian, DF, at the given point X and increments NDV by N^2 for the number of function evaluations. N is the dimension.

The fifth entry, NFEV(NE), line 36, sets NE to the final number of function evaluations and resets the counters, NF and NDV.

The sixth entry, FHEAD(AL,INDF,N,NX,XX), line 42, gives the necessary information about the system of equations. INDF is set to the file index number, N is set to the dimension, the 26 words of AL are set to a heading, NX is set to the number of solutions given in XX, and finally, XX(I,J); I = 1,N; J = 1,NX is set to the solutions.

4.7 Initial-Value Generating Routine

A properly programmed Initial-Value Generating Routine is illustrated on page A-8. The Fortran control statement (line 1) uses the file index number (1 in this example) as a version. The first entry INVAL(*,X), line 2, is used to return a new initial value in X upon subsequent calling. A return to * occurs when all initial values generated by the routine were given before. The second entry INVIND(INDS), line 26, is used to return the file index number in INDS.

The presented example is a type of Initial-Value Generating Routine which we adopted for 2-dimensional cases. The generated initial values are on concentric circles. The origin of the circles is returned by using

the third entry (PARM(M,Z), line 29. The initial value points on one circle are equally distributed, but the circles can be rotated so that the corresponding points on the different circles will lie on a spiral. The parameters are listed in a Data Statement (line 11). By changing this statement, it is easy to obtain another routine for the 2-dimensional case; naturally, the new routine should have a different file index.

5. List of Tape-Output File Program

This program lists the contents of the Tape-Output File (File 2). For the outputs of each run by the Method Testing Program, the List program prints one line of output. An example is shown on page A-9. The notation for the Initial-Value File is as follows: the first integer is the index number of the Initial-Value Generating Program. It is followed by the letter S. The next two integers indicate the numbering of the initial values in the set generated by the routine, i.e., 1S1-40 , indicates the first Initial-Value Generating program which generated forty different initial values numbered from 1 to 40.

Deck Set-up:

```
'RUN        ...
'ASG,T      PR,8C,t1
'ASG,T      2,8C,t2
'ASG,T      A,F
'COPIN      PR,A
'FREE       PR
'XQT        A,LISTX
'FIN
```

6. Table Generating Program

The Table Generating Program selects data from the Tape Output file and arranges them in a table form. This printed table provides a quick, visual comparison of different methods. A sample output is shown on page A-10.

There are three types of selections to be defined by two data cards:

1. The basic selection defines, for all iterations to be included in the table, the indices of the function file and of the initial-value file, the maximum allowable number of iterations, the norm, and the number i_0 of consecutive terms used in the convergence and divergence criteria.

2. The method selection defines the methods by name. A maximum of nine methods can be used for one table, and they define the columns of the table. The rows of the table are defined by the different initial values in the set of initial values defined by the basic selection.

3. The result selection defines the data to be printed in the table. These always consist of the type of return and nothing else or one of the following: the number of iterations, the number of function evaluations, the time spent in msec, and the solution index.

The format of the two data cards is as follows:

First card:	col.	1 - 5	Function file index
	"	6 - 10	Initial Value file index
	"	11 - 15	Maximum number of iterations
	"	20 - 25	Name of the norm
	"	31 - 35	i_0
	"	36 - 40	ND

```
Second card:  col.  1 -  5     Number of methods to be listed
               "    8 - 13     Name of the first method
               "   16 - 21       "   "  "  second  "
               "   24 - 29       "   "  "  third   "
                     .
                     .
                     .
                   up to
               "   72 - 77       "   "  "  ninth   "
```

where ND is defined as the result selection:

```
     if  ND = 1  only the type of return is to be printed
              2  type of return and the number of iteration
              3    "  "    "     "  the solution index
              4    "  "    "     "  the number of function evaluations
              5    "  "    "     "  the time spent in msec
```

Deck Set-up:

```
'RUN          ...
'ASG,T        PR,8C,t1
'ASG,T        2,8C,t2
'ASG,T        A,F
'COPIN        PR,A
'FREE         PR
'XQT          A.TBX
 (set of 2 data cards)
'FIN
```

```
MAX. NUMBER OF ITERATIONS =    50
PRINT INDEX               =     1
PUNCH INDEX               =     3
ACCURACY REQUIREMENTS     =  .1-06  .1-06  .1-05  .5+01
NORM                      =  L-TWO
INITIAL VALUE FILE        =     1S
DIMENSION                 =     2
FUNCTION FILE             =     1F
        (1F)    F(1) = X(1)**2 + X(2)**2 - 1.0
                F(2) = X(1)**3 - X(2) - 1.0
     3 KNOWN SOLUTIONS    =
     1       .10000000+01        .00000000
     2       .00000000          -.10000000+01
     3       .54368901-00       -.83928675-00
```

```
                1-TH INITIAL VALUES
      .11000000+01              .00000000
 0       .39-00          .11000000+01          .00000000
 1       .31-01          .10045455+01         -.15500005-01
 2       .28-03          .10001352+01          .34720078-03
 3       .17-06          .10000001+01          .15390106-06
 4       .54-07          .10000000+01          .48894968-08
 5       .54-07          .10000000+01          .48894962-08

RETURN WITH C       SOLUTION =  1
NUMBER OF FUNCTION EVALUATION =    32     TIME =     32 MS.

METHOD NEWTON,     FUNCTION     1,     IN.VAL.     1,     NORM L-TWO
```

```
          15-TH INITIAL VALUES
     .12876553+01          -.52654954-00

 0       .19+01          .12876553+01          -.52654954-00
 1       .35-00          .98180611-00          -.38634059-00
 2       .19+01          .45050039-00          -.15900403+01
 3       .39-00          .29373206-01          -.11649747+01
 4       .32-00          .53041797-00          -.99867779-00
 5       .16-01          .50366097-00          -.87335421-00
 6       .45-02          .54858996-00          -.83804213-00
 7       .45-04          .54378632-00          -.83923832-00
 8       .33-07          .54368909-00          -.83928672-00
 9       .00             .54368901-00          -.83928677-00

RETURN WITH C       SOLUTION =  3
NUMBER OF FUNCTION EVALUATION =   56      TIME =     53 MS.

METHOD NEWTON,      FUNCTION      1,      IN.VAL.      1,      NORM L-TWO
```

4	SECANT	1F	1S	1	C	1	6	18	42	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	2	C	1	6	18	51	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	3	C	1	7	20	56	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	4	C	1	6	18	51	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	5	C	1	5	16	45	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	6	C	1	5	16	36	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	7	C	1	4	14	29	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	8	C	1	6	18	47	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	9	C	1	14	34	92	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	10	C	1	8	22	54	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	11	C	1	9	24	60	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	12	C	1	9	24	69	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	13	C	1	15	36	99	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	14	C	3	11	28	73	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	15	C	1	14	34	92	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	16	D	0	9	24	64	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	17	C	1	12	30	79	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	18	C	1	9	24	61	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	19	D	0	21	48	133	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	20	CB	0	8	22	60	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	21	C	3	8	22	65	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	22	C	3	29	64	187	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	23	C	1	12	30	77	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	24	C	3	14	34	89	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	25	C	1	10	26	69	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	26	D	0	44	94	297	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	27	C	3	17	40	132	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	28	C	0	18	42	138	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	29	C	3	8	22	66	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	30	C	1	11	28	84	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	31	CB	0	7	20	49	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	32	C	3	14	34	92	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	33	D	0	14	34	91	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	34	CB	0	46	98	292	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	35	I	0	50	106	336	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	36	C	0	27	60	181	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	37	C	3	20	46	131	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	38	BC	0	30	66	187	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	39	C	3	17	40	122	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECANT	1F	1S	40	C	3	12	30	81	L-TWO	50	.1-06	.1-06	.1-05	5

```
'FOR,ST   A.LTWO,A.LTWOX                                                1
      FUNCTION ENORM(X,N)                                               2
      DIMENSION X(1)                                                    3
      S=0.                                                              4
      DO 5 I=1,N                                                        5
      IF (ABS(X(I)).GT.S) S=ABS(X(I))                                   6
    5 CONTINUE                                                          7
      T=0.                                                              8
      IF (S.EQ.0.) GO TO 15                                             9
      DO 10 I=1,N                                                      10
   10 T=T+(X(I)/S)**2                                                  11
      T=S*SQRT(T)                                                      12
   15 ENORM=T                                                          13
      RETURN                                                           14
C                                                                      15
      ENTRY ENORMH(EE)                                                 16
      DATA ENM/5HL-TWO/                                                17
      ENORM=ENM                                                        18
      RETURN                                                           19
      END                                                              20
```

```
'FOR,SI  A.METHOD/NEWTON,A.METHDX/NEWTON                                    1
      SUBROUTINE METHOD(*,*,*,N,X,MAX,ACC,IT,AN)                            2
C               NEWTON METHOD                                               3
      DIMENSION X(25),ACC(2),DX(25),DF(25,25),F(25)                         4
      DATA ANE/6HNEWTON/                                                    5
C***                                                                        6
C                                                                           7
      IT=0                                                                  8
      ICC=0                                                                 9
      IDC1=0                                                               10
      IDC2=0                                                               11
      I0=ACC(4)                                                            12
      CALL OUTPUT(IT,X,N)                                                  13
      AN=ANE                                                               14
C                                                                          15
C***                                                                       16
      CALL FX(F,X)                                                         17
      FN=FNOR      I)                                                      18
C                                                                          19
   10 IT=IT+1                                                              20
      CALL DERIVF(X,DF)                                                    21
      CALL GSLV($450,DF,F,N)                                               22
      DO 20 I=1,N                                                          23
      DX(I)=F(I)                                                           24
   20 X(I)=X(I)-DX(I)                                                      25
      CALL FX(F,X)                                                         26
C***                                                                       27
      CALL OUTPUT(IT,X,N)                                                  28
      XN=ENORM(X,N)                                                        29
      DXXN=ENORM(DX,N)                                                     30
      IF (DXXN.LE.ACC(2)) RETURN                                           31
      FNN=ENORM(F,N)                                                       32
      IF ((XN.GE.1.E20).OR.(FNN.GE.1.E20)) RETURN 1                        33
      IF (IT.LE.1) GO TO 400                                               34
      IF (DXXN.GE.DXN) ICC=0                                               35
      IF (ICC.LT.I0) ICC=ICC+1                                             36
      IF (DXXN.LT.DXN) IDC1=0                                              37
      IF (IDC1.LT.I0) IDC1=IDC1+1                                          38
      IF (FNN.LE.FN) IDC2=0                                                39
      IF (IDC2.LT.I0) IDC2=IDC2+1                                          40
      IF (IT.LE.I0) GO TO 400                                              41
      BETA=XN*ACC(3)                                                       42
      IF (BETA.LT.ACC(3)) BETA=ACC(3)                                      43
      IF ((DXXN.LE.BETA).AND.(ICC.EQ.I0)) RETURN                           44
      IF (IDC1.GE.I0) RETURN 1                                             45
      IF ((IDC2.GE.I0).AND.(ICC.LE.1)) RETURN 1                            46
  400 DXN=DXXN                                                             47
      FN=FNN                                                               48
      IF (IT.GE.MAX) RETURN 2                                              49
C***                                                                       50
      GO TO 10                                                             51
  450 RETURN 3                                                             52
      END                                                                  53
```

```
'FOR,SI  A,F/1,A,FX/1
      FUNCTION FXK(K,X)
      DIMENSION X(1)
      EQUIVALENCE (NF,FN)
      DATA NF/0/
      NF=NF+1
      GO TO (1,2),K
    1 FXK =X(1)**2+X(2)**2-1.0
      RETURN
    2 FXK =X(1)**3-X(2)-1.0
      RETURN
C
      ENTRY NFEK(K)
      FXK=FN
      NF=0
      RETURN
C
      END
'FOR,SI  A,D/1,A,DX/1
      SUBROUTINE FX(F,X)
      DIMENSION F(1),X(1)
      DO 10 J=1,2
   10 F(J)=FXK(J,X)
      RETURN
C
      ENTRY DERIVF(X,DF)
      DIMENSION DF(25,25)
      DATA NDV/0/
      NDV=NDV+4
      DF(1,1)=2.*X(1)
      DF(1,2)=2.*X(2)
      DF(2,1)=3.*X(1)**2
      DF(2,2)=-1.0
      RETURN
C
      ENTRY NFEV(NF)
      NF=NFEK(0)
      NF=NF+NDV
      NDV=0
      RETURN
C
      ENTRY FHEAD(AL,INDF,N,NX,XX)
      DIMENSION AL(26),XX(26,25),A(26)
      DATA (A(I),I=1,26)/78H     (1F)     F(1) = X(1)**2 + X(2)**2 - 1.0
     1                                    ,78H              F(2) = X(1)**3 -
     2 X(2) - 1.0                                          /
      INDF=1
      N=2
      NX=3
      XX(1,1)=1.0
      XX(2,1)=0.0
      XX(1,2)=0.0
      XX(2,2)=-1.0
      XX(1,3)=0.543689013
      XX(2,3)=-.839286755
      DO 50 I=1,26
   50 AL(I)=A(I)
      RETURN
C
      END
```

```
!FOR,SI  A,S/1,A,SX/1                                                    1
      SUBROUTINE INVAL(*,X)                                              2
      DIMENSION X(1)                                                     3
      DATA IR,IM/0,0/                                                    4
      DATA P2/6.2831853/                                                 5
C     STARTING VALUES ON CONCENTRIC CIRCLES                              6
C        X0,Y0 = ORIGIN                                                  7
C        R0,DR = STARTING RADIUS AND RADIUS FACTOR                       8
C        NR    = NUMBER OF CIRCLES                                       9
C        MN,MD = STARTING NUMBER OF POINTS ON (R0) AND INCREMENT        10
      DATA X0,Y0,R0,DR,NR,MN,MD,AL,DA/1.0,0.,.1,.5,5,8,0,0.,.5/          11
C                                                                        12
      IF (IR.EQ.NR) RETURN 1                                             13
      AE=AL+P2*FLOAT(IM)/FLOAT(MN)                                       14
      X(1)=X0+R0*COS(AE)                                                 15
      X(2)=Y0+R0*SIN(AE)                                                 16
      IM=IM+1                                                            17
      IF (IM.LT.MN) RETURN                                               18
      IR=IR+1                                                            19
      R0=R0+DR                                                           20
      IM=0                                                               21
      MN=MN+MD                                                           22
      AL=AL+DA                                                           23
      IF (AL.GT.P2) AL=AL-P2                                             24
      RETURN                                                             25
      ENTRY INVIND(INDS)                                                 26
      INDS=1                                                             27
      RETURN                                                             28
      ENTRY PARM(M,Z)                                                    29
      DIMENSION Z(1)                                                     30
      M=2                                                                31
      Z(1)=X0                                                            32
      Z(2)=Y0                                                            33
      RETURN                                                             34
      END                                                                35
```

LIST OF CONTENTS

	METHOD	F-FILE	IN.VAL.FILE	NORM	MAX	EPS1	EPS2	EPS3	IO
1	NEWTON	1F	1S1- 40	L-TWO	50	.1-06	.1-06	.1-05	5
2	MODNWT	1F	1S1- 40	L-TWO	50	.1-06	.1-06	.1-05	5
3	SECANT	1F	1S1- 40	L-TWO	50	.1-06	.1-06	.1-05	5
4	SECNTI	1F	1S1- 40	L-TWO	50	.1-06	.1-06	.1-05	5
5	SECNTP	1F	1S1- 40	L-TWO	50	.1-06	.1-06	.1-05	5
6	A315	1F	1S1- 40	L-TWO	50	.1-06	.1-06	.1-05	5
7	A316	1F	1S1- 40	L-TWO	50	.1-06	.1-06	.1-05	5

T A B L E

NUMBER OF ITERATIONS - TYPE OF RETURN

FUNCTION FILE = 1

INITIAL VALUE FILE = 1

MAX.NUMBER OF ITER.= 50

NORM = L-TWO

IO = 5

IN.VAL.	NEWTON	MODNWT	SECANT	SECNTI	SECNTP	A315
1	5-C	6-C	6-C	6-C	6-C	5-C
2	4-C	7-C	6-C	6-C	6-C	4-C
3	4-C	9-C	7-C	7-C	7-C	4-C
4	4-C	7-C	6-C	6-C	6-C	4-C
5	5-C	7-C	5-C	5-C	5-C	5-C
6	4-C	8-C	5-C	5-C	5-C	4-C
7	5-C	15-C	4-C	4-C	5-C	5-C
8	4-C	10-C	6-C	6-C	6-C	4-C
9	6-C	24-CB	14-C	14-C	14-C	6-C
10	6-C	24-C	8-C	8-C	8-C	6-C
11	6-C	50-I	9-C	9-C	8-C	6-C
12	8-C	6-D	9-C	9-C	12-C	5-C
13	8-C	16-D	15-C	15-C	10-C	6-C
14	6-C	8-D	11-C	11-C	9-C	6-C
15	9-C	50-I	14-C	14-C	12-C	8-C
16	19-C	4-D	9-D	9-D	35-C	8-C
17	6-C	35-CB	12-C	12-C	11-C	6-C
18	6-C	45-CB	9-C	9-C	9-C	6-C
19	13-C	3-D	21-D	21-D	50-I	6-C
20	24-C	3-D	8-CB	8-CB	23-BC	17-C
21	7-C	5-D	8-C	8-C	7-C	6-C
22	7-C	50-I	29-C	27-C	12-C	7-C
23	6-C	6-D	12-C	12-C	12-C	6-C
24	9-C	6-D	14-C	14-C	13-C	9-C
25	7-C	50-I	10-C	10-C	12-C	7-C
26	38-C	2-D	44-D	44-D	50-I	6-D
27	18-C	5-D	17-C	17-C	27-BC	17-C
28	13-C	50-I	18-C	13-C	18-CB	13-C
29	6-C	50-I	8-C	8-C	9-C	6-C
30	6-C	10-D	11-C	11-C	11-C	6-C
31	17-C	3-D	7-CB	7-CB	36-C	9-C
32	9-C	11-D	14-C	14-C	14-C	9-C
33	18-C	3-D	14-D	14-D	24-C	8-C
34	11-C	6-D	46-CB	47-CB	12-C	6-D
35	16-C	50-I	50-I	24-BC	29-BC	16-C
36	20-C	3-D	27-C	24-C	28-C	17-C
37	8-C	50-I	20-C	18-C	27-C	8-C
38	8-C	6-D	30-BC	27-BC	23-C	8-C
39	7-C	6-D	17-C	17-C	17-C	7-C
40	7-C	45-CB	12-C	12-C	12-C	7-C